

1. **PART I**

| **1. Personal Background** |
| --- |
| . |

| Students’ Name | * Branco Javier Molina Ulloa * Lucas Matias Antil Gaete * Cristobal Ignacio Cona Cona * Joan Garcés Maldonado |
| --- | --- |
| National ID (RUT) | * 20.282.338-6 * 21.457.145-5 * 20.634.537-3 * 20.938.288-1 |
| Degree Program | Computer Engineering |
| Campus | Puente Alto |

| **2. APT Project Description** |
| --- |
|  |

| Project Name: | **Meetings APP** |
| --- | --- |
| Area(s) of Specialization: | Web development and programming; Software architecture; Databases; Cloud/AI service integration; Quality assurance; Project management. |
| Competencies | .Gather, analyze, and prioritize functional and non-functional requirements.  Design the architecture of a secure, scalable, and maintainable web solution.  Develop front-end and back-end components, integrating APIs (e.g., speech-to-text/translation) and third-party services (e.g., payments/subscriptions).  Model and implement the database and ensure data integrity.  Automate tests and apply usability and quality standards.  Manage the project lifecycle through planning, change control, and documentation. |

| **3. Rationale – APT Project** |
| --- |
|  |

| APT Project Relevance | Meetings Inc. is currently in a key expansion phase within the tech sector, focusing its operations on facilitating multilingual virtual meetings for clients across various industries. Its main users include corporations, consultants, universities, and professionals operating in globalized environments where language remains a persistent barrier to effective communication.  This project emerges from the lack of video conferencing tools that efficiently integrate features such as real-time translation, automatic subtitles, and AI-generated meeting summaries. Many organizations are forced to hire external interpreters, use multiple platforms simultaneously, or incur high costs due to fragmented software solutions.  Meetings APP addresses this challenge by offering a professional web solution that centralizes video conferencing, real-time translation, and reporting in a single environment. This is especially relevant in today's workforce, enhancing inclusion, decision-making, and cost-efficiency. It benefits not only large enterprises but also SMEs and freelancers needing barrier-free communication.  Headquartered in Puente Alto, Santiago, the company aims to position itself as a Chilean tech platform that balances usability, accessibility, and security. Project Manager Branco Molina and his multidisciplinary team lead the development with a strategic, future-oriented vision, leveraging tools such as Microsoft Azure Speech, machine translation services, and payment APIs like MercadoPago or Transbank—ensuring scalability and interoperability.  The main beneficiaries will be professionals and organizations working in international environments, while those most affected by the current lack of integration are losing time, budget, and operational efficiency. |
| --- | --- |
| APT Project Description | Meetings is a web-based videoconferencing platform tailored for SMEs, institutions, and work teams. It allows for:   * Meeting creation and management with audio/video. * Real-time automatic subtitles and translation, post-meeting transcription, and summaries. * User and subscription management (free and paid plans). * Admin dashboard with usage metrics and reporting. * Privacy and security controls (roles/permissions, responsible data handling). |
| Alignment with Graduate Profile | The Meetings App project is fully aligned with the Computer Engineering graduate profile at Duoc UC, enabling practical application of multiple core competencies.  It begins with requirement analysis, evidenced in the ERS document, showing deep understanding of client needs. Technically, it involves designing and developing modern computing solutions using technologies like React, Django, and Azure AI Speech, integrating automatic translation, payment gateways, and strong authentication.  Project management follows PMI best practices: stage-based planning, WBS estimation, Gantt tracking, and change control. Information security is addressed through encryption, role-based access, and GDPR compliance.  The project also fosters collaborative work, applied innovation, and continuous adaptation to emerging technologies—especially in AI and multilingual communication. |
| Connection to Professional Interests | Our professional interests lie in full-system software development, particularly interface design, web programming, cloud service integration, security, and databases.  The Meetings App reflects these interests, allowing us to apply our technical skills in a realistic context to solve the very real challenge of multilingual communication in digital environments. |
| APT Project Feasibility | The project will be executed over the academic semester (approximately 5 months), with a weekly commitment of 5 hours. This allows the team to organize the work into phases and meet deadlines for required deliverables (project charter, ERS, DAS, development, testing, etc.).  **Technology Stack:**   * **Frontend:** React with Tailwind CSS for responsive and intuitive UI. * **Backend:** Django (Python), layered architecture. * **Cloud Services:** Microsoft Azure for hosting, processing, and storage. * **Real-time Translation:** Azure AI Speech for speech-to-text and live subtitle generation. * **Video Calls & Chat:** WebRTC, Socket.IO, and WebSocket for low-latency real-time communication. * **Database:** Oracle SQL Developer 19c.   The tech stack has been reviewed and is familiar to the team, facilitating technical development. There are no external limitations to progress, as the project operates in a controlled but professionally realistic environment.  **Risks Identified:**   * High dependency on external services (e.g., Azure AI Speech, MercadoPago) which could affect critical functionality. * Technical complexity in integrating live video and subtitle synchronization. * Academic/personal workload potentially causing delays if weekly hours aren't met. * General risks like technical failures, internet issues, or key member absences. |

1. **PART II**

| **4. Objectives** |
| --- |
|  |

| General Objective | Develop a professional-grade video conferencing platform that enables multilingual communication through real-time translation, automatic subtitles, and meeting summary generation. |
| --- | --- |
| Specific Objectives | Implement real-time translation with subtitle overlay.  Develop a module to generate automated meeting summaries for subscribed users.  Integrate video call and chat functionalities using WebRTC and WebSocket.  Create maintenance tools for users, profiles, rooms, languages, subscriptions, and system settings.  Design a structured and optimized database to manage system information.  Ensure data security and confidentiality through authentication, roles, and encryption.  Integrate a payment gateway for subscription management.  Guarantee cross-platform compatibility and responsive UI design. |

| **5. Methodology** |
| --- |
|  |

| Methodology Description |
| --- |
| A hybrid methodology will be used, combining classical project management (PMI) with agile practices to structure project stages and adapt to changes during development.  **From the classical approach:**  Sequential phases with key deliverables (charter, ERS, DAS, development, testing, closure) will be followed. Each phase will include planning, assigned roles, tools, and change control.  **From the agile approach:**  Iterative cycles will prioritize core functionalities like real-time subtitles, translation API integration, and user/subscription modules. Weekly reviews will be held, with progressive improvements.  **Main Project Stages:**   * Initial definition and documentation (charter, ERS, requirements, mockups). * Software architecture and database design. * Iterative development of core features. * Integration of external services (translation, payment gateway). * Execution of quality, security, and performance tests. * Final delivery, documentation, and product presentation.   Throughout, a layered architecture will be applied, clearly separating presentation, business logic, and data access—using React, Django, WebSocket, WebRTC, and Azure services. |

| **6. Evidence** |
| --- |
|  |

| **Type** | **Evidence Name** | **Description** | **Justification** |
| --- | --- | --- | --- |
| Progress | Initial Project Presentation | A general project document explaining the context, problem, solution, and team | Introduces the project to stakeholders and validates the initial approach |
| Progress | Project Charter | A formal document defining the project's scope, objectives, deliverables, and team | Serves as the foundational framework for project management (PMI methodology) |
| Progress | Software Requirements Specification (SRS) | Document defining functional and non-functional requirements | Essential for system design, development, and validation |
| Progress | Requirements Spreadsheet | Detailed matrix of functional and non-functional requirements | Enables tracking, validation, and visualization of agreed requirements |
| Progress | Gantt Chart | Visual timeline of project activities, milestones, and deadlines | Facilitates monitoring of project progress and deadlines |
| Progress | Work Breakdown Structure (WBS) | Structured task hierarchy with estimated effort per activity | Aids in workload distribution and planning across the team |
| Progress | Extended Use Cases | Detailed description of system-user interactions | Helps identify critical features and key scenarios |
| Progress | System Mockups | Visual prototype in Figma illustrating the interface and screen flow   |  | | --- |  |  | | --- | | Allows for early validation by users and stakeholders |
| Progress | |  | | --- |  | Phase 1 Self-Evaluation | | --- | | Individual assessment of personal performance during Phase 1 | Encourages reflection on individual contributions and learning |
| Progress | APT Project Definition – Phase 1 | Guiding document consolidating initial planning and methodology | Provides a structured foundation for later stages of the project |
| Progress | Business Process Document (TO-BE) | Describes the improved business process post-implementation | Demonstrates the impact of the system compared to current workflows |
| Progress | RACI Matrix | Defines roles and responsibilities within the team | Prevents role confusion and enhances coordination |
| Progress | |  | | --- |  | Risk Matrix | | --- | | Identifies potential risks and mitigation strategies | Helps anticipate problems and reduce their impact |
| Progress | Data Dictionary | Defines entities, attributes, and relationships in the system | Standardizes technical language across the team |
| Progress | Change Control Log | Records changes made throughout the project | Ensures traceability and accountability of system updates |
| Progress | |  | | --- |  | Phase 2 Project Progress Report | | --- | | Report detailing technical and functional progress of Phase 2 | Assesses partial completion and alignment with objectives |
| Final | Final Project Report | Comprehensive summary of project work and results | Demonstrates objective achievement and key takeaways |
| Final | 100% System Development | Fully functional codebase of the web and desktop apps with integrated features | Represents the final product fulfilling client requirements |
| Final | |  | | --- |  | Database Script | | --- | | Scripts for table creation, relationships, and initial inserts | Allows execution of the proposed database model |
| Final | |  | | --- |  | Testing Plan | | --- | | Documentation of tests performed, results, and observations | Ensures system compliance with defined quality criteria |
| Final | Database Test Matrix | Validates functionality of database queries and structures | Ensures database performance and integrity |
| Final | Defect Log | Detailed log of identified bugs and resolutions | Tracks quality control and system improvements during testing |
| Final | |  | | --- |  | Scope Verification | | --- | | |  | | --- |  | Assesses whether initial project requirements were met | | --- | | Confirms project compliance with original planning |
| Final | Training Plan | Training proposal for end-users | Facilitates user adoption and system onboarding |
| Final | User Manual | Document guiding proper use of the application   |  | | --- |  |  | | --- | | Helps users understand and maximize the system’s capabilities |
| Final | Project Closure Report   |  | | --- |  |  | | --- | | Final document formalizing project completion (admin + technical) | Marks the official end of the project with proper documentation |

| **7. Work Plan** |
| --- |
|  |

| **APT Project Work Plan** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Competency Area | Task Name | Description | Recursos | Duration | Responsible | Observations |
| Effective communication, PM | Initial Project Presentation | Prepare and deliver general project overview (context, problem, solution) | PC, guide documents, projector | 11–18 Aug 2025 | Entire Team | |  | | --- |  | Base document. Deliverable: Initial Project Presentation | | --- | |
| Planning, Technical Writing | Charter, Gantt Chart, Requirements Sheet | Draft the formal framework, timeline, and requirements | Charter template, MS Project, Excel | 18–25 Aug 2025 | PM / Architect | Deliverables: Charter, Gantt Chart, Requirements Sheet |
| Planning, Documentation | ERS, Mockups, WBS Matrix | Write IEEE 830-standard ERS, design Figma mockups, build WBS | ERS doc, Figma, Excel | 25 Aug – 1 Sep 2025 | Architect / Frontend Dev / PM | Deliverables: ERS Report, Mockups, WBS Matrix |
| Process Modeling, Self-mgmt | APT Definition, Phase 1 Eval, Use Case | Consolidate project scope, perform self-evaluation, write extended use case | Course guides, use case docs | 1–8 Sep 2025 | Entire Team / Analyst | Deliverables: APT Definition, Phase 1 Eval, Use Case Document |
| Software Architecture | Architectural Design | Define layered architecture, TO-BE model, and diagrams | Modeling tools | 16–30 Sep 2025 | Software Architect | Deliverable: DAS (Architectural Design Document) |
| Programming, DB Management | Backend Development | Implement server logic, database, and APIs | Django, Azure, GitHub | 5 Sep – 7 Oct 2025 | Backend Developer | Deliverables: Backend Code, DB Script |
| Web Dev, Integration | Frontend Development | Build web UI and integrate backend + translation API | React, Figma, GitHub | 5 Sep – 7 Oct 2025 | Frontend Developer | Deliverable: Frontend Code |
| Programming, Architecture | Admin Desktop App | Build desktop app for system maintainers | .NET / Electron | 20 Sep – 7 Oct 2025 | Architect / Backend Dev | Deliverable: Desktop App Code |
| Service Integration, Security | External Integrations | Configure Azure AI Speech + payment gateway | Azure AI, External APIs | 25 Sep – 15 Oct 2025 | Backend / CISO | Deliverable: Functional Integrations |
| Testing, QA | Unit Testing | Validate isolated components | Postman, Jest, PyTest | 15–30 Oct 2025 | QA / CISO | Deliverable: Unit Testing Report |
| Cybersecurity, QA | Integration & Security Testing | Test module cohesion, role handling, and system security | Selenium, OWASP, Azure | 17–25 Nov 2025 | QA / CISO | Deliverables: QA Report, Defect Matrix |
| Teamwork, Communication   |  | | --- |  |  | | --- | | Final Delivery & Closure | Present system, deploy final version, and formally close the project | Classroom, docs, live demo | 26 Nov – 5 Dec 2025 | Entire Team | Deliverables: Final Report, Final Presentation |

| **8. Gantt Chart** |
| --- |
|  |

**Link to Gantt Chart (MPP Project File):** <https://drive.google.com/file/d/1Po2WLglD6F3hNgqyb_TfWbON9Lyp28AC/view?usp=drive_link>

